Segmentador de oraciones para textos en español basado en red neuronal

José I. Espinosa, José C. González, Amalio F. Nieto y José M. Goñi
E.T.S.I. Telecomunicación
Universidad Politécnica de Madrid
28040 Madrid
E-mail: {jiespino,jcg,anieto,jmg}@gsi.dit.upm.es

Resumen

En este trabajo se muestra la utilización de una red neuronal para segmentar textos en español en sus oraciones constitutivas. Esta operación debe efectuarse habitualmente como un paso previo en multitud de aplicaciones de procesamiento de lenguaje natural. A pesar de tratarse de una tarea conceptualmente sencilla, y de obtenerse resultados aceptables por diversos procedimientos, la división de un texto en oraciones tiene el inconveniente de ser fuertemente dependiente de la fuente (estructura, tipo de lenguaje, género literario, etc.) Esto obliga prácticamente a rehacer el trabajo no sólo para cada tipo de aplicación, sino para cada tipo de texto que vaya a ser tratado. Frente a otros tipos de técnicas, la utilización de redes neuronales tiene la ventaja de liberar al desarrollador de la tarea de programación, empleándose colecciones de ejemplos correctamente clasificados para el entrenamiento del segmentador.

Abstract

This work shows a connectionist system used for the segmentation of texts in Spanish into separate sentences. This task has to be carried out in many Natural Language Processing applications. This kind of pre-processing is not conceptually complex, and several techniques producing acceptable results may be applied. However, the task of text segmentation depends heavily on the sources (structure, layout, genre, style). Most of times, this fact imposes some reworking for every new application and type of text. By using neural nets, low level programming is replaced by learning from sets of sentences correctly classified by a specialist.

1 Introducción

El presente trabajo partía de una doble motivación. Un primer objetivo consistía en resolver de manera cómoda el problema de la división de textos extensos, dotados de una estructura supra-oracional, en sus oraciones constitutivas. El segundo móvil era explorar la utilización de redes neuronales en el campo del procesamiento de lenguaje natural.

En cuanto al problema abordado, los puntos no sirven sólo como delimitadores entre oraciones. Pueden aparecer al final de abreviaturas (Sr., Dr., Vd., Ilmo.), al final o entre medias en acrónimos (O.N.U., F.B.I., M.I.R.) o pueden pertenecer a una abreviatura o acrónimo y

ser a la vez final de oración. Esta multiplicidad en la función que pueden desempeñar genera una ambigüedad que es la hemos tratado de resolver.

Los métodos conexionistas [2] han atraído recientemente la atención de buen número de investigadores en diversas áreas de la ingeniería lingüística, como demuestra la existencia de varias monografías sobre el tema [1, 5, 7, 8]. El problema concreto de la división de un texto en oraciones mediante redes neuronales ya ha sido tratado con éxito para el idioma inglés [3, 4]. Los métodos conexionistas aquí empleados son los descritos en estos dos trabajos.

2 Métodos convencionales de segmentación

En esta sección describiremos brevemente algunos de los métodos que se han utilizado para dividir textos en oraciones para idiomas como el inglés o el alemán. (Lamentablemente, no conocemos referencias publicadas de trabajos similares para el español).

- **Expresiones regulares y reglas heurísticas**
  Este método consiste en usar una gramática regular, normalmente con algo de conocimiento de las palabras anteriores o posteriores a los posibles finales de oración. La forma más sencilla de realizar este método es crear reglas que busquen ciertos patrones de caracteres. Aunque estas aproximaciones pueden ser muy efectivas, requieren un importante esfuerzo manual para generar las reglas que reconocen los límites de oración. Tales esfuerzos son normalmente específicos del conjunto de textos que se está estudiando, por lo que probablemente no serán aplicables a otros géneros literarios. Además, dado que se fundamentan en listas de palabras específicas de un lenguaje, no son exportables a otros lenguajes naturales sin repetir el esfuerzo de generar las listas de palabras y las reglas. Por último, estas aproximaciones heurísticas se basan en que el conjunto de textos sobre los que trabajan sean bastante regulares en cuestión de puntuación y tengan pocos caracteres extraños. Probablemente no darán buenos resultados si trabajan sobre textos obtenidos vía reconocimiento óptico de caracteres (*scanners*).

- **Árboles de regresión**
  El método utiliza información sobre el contexto formado por la palabra anterior y posterior al signo de puntuación. Debe guardar, para cada palabra del léxico, la probabilidad de que ocurra en las proximidades de un límite de oración. El éxito del método es muy alto, pero precisa una enorme carga de almacenamiento.

- **Finales de palabra y listas de palabras**
  Se intenta identificar abreviaturas y palabras que rodean a los signos de puntuación. El método se basa en realizar múltiples pasadas sobre los datos para encontrar sufijos reconocibles y poder filtrar así las palabras que no es probable que sean abreviaturas. El análisis morfológico permite identificar palabras que no están presentes en las grandes listas de palabras que utiliza para identificar abreviaturas. Adolece también de los problemas que presentaba el primer método.

3 Un enfoque conexionista

3.1 Descripción del sistema

El objetivo de este trabajo es lograr resolver las posibles ambigüedades en los signos de puntuación que pueden actuar como finales de oración en un texto en castellano. Vamos a utilizar la red neuronal para resolver dicha ambigüedad. Lo que hacemos es tomar cada posible límite

de oración que aparece en un texto y analizar si lo es realmente (o si por el contrario es, un punto de abreviatura, de un acrónimo, parte de un número, etc). Para ello tomaremos un contexto alrededor del posible fin de oración. Para cada una de las palabras que componen ese contexto tomaremos la probabilidad de que pertenezcan a una clase gramatical u otra (esto es, si es nombre, verbo...). Son esas probabilidades las que introduciremos a la red neuronal como entrada. La salida de la red nos indicará si el signo de puntuación era fin de oración o no. En líneas generales el proceso que hemos desarrollado consiste en preparar adecuadamente el texto de entrada, de modo que sea manejable por el simulador de redes neuronales. Para ello se descompondrá en unidades (*tokens*) y se tratará convenientemente cada una de esas unidades. Cuando se ha hecho este proceso sólo resta que la red neuronal clasifique los signos de puntuación como finales de oración si realmente lo son. Para que el resultado sea fácilmente accesible, la salida que proporciona la red es tratada de nuevo, de modo que se puede comprobar cómo ha quedado descompuesto el texto original en las oraciones que lo forman.

3.2 La red neuronal

El papel de la red neuronal es resolver la ambigüedad de la que hemos hablado. Para ello hemos escogido (siguiendo a Palmer y Hearst) una red neuronal alimentada hacia adelante (feed-forward). Concretamente, se ha tomado un perceptrón de tres capas, en el que la capa de entrada cuenta con 138 nodos, la oculta con dos y con uno la capa de salida. El algoritmo de aprendizaje es *back propagation.*

3.3 Descripción global del sistema

Como se ha indicado anteriormente el proceso comienza tomando cada palabra o signo del texto e identificando los signos que pueden ser finales de oración. Para cada uno de estos signos se toma un contexto. Este contexto lo escogimos de tres palabras anteriores y tres posteriores. Para cada una de ellas se toma la probabilidad de que pertenezca a cada una de las clases gramaticales que identificamos para el castellano (21 en total). Estas probabilidades han sido identificadas a priori sobre un léxico que creamos a propósito para este sistema. El léxico fue extraído de la colección de textos publicada por el diario *ABC* en su suplemento cultural. Para generarlo se realizaron numerosos procesos de filtrado, así como un exhaustivo etiquetado de cada una de las palabras y signos del mismo. Este etiquetado se realiza mediante un etiquetador estadístico desarrollado en colaboración con el Laboratorio de Lingüística informática de la Universidad Autónoma de Madrid y financiado por la Unión Europea a través del proyecto CRATER (MLAP93-20) [6].

Una vez etiquetados los textos, se calcularon las frecuencias de aparición de cada unidad (palabra o signo) bajo cada una de las posibles categorías gramaticales. Entonces se calcularon las probabilidades asociadas a cada palabra con cada categoría, de modo que formamos un vector con dichas probabilidades. Así pues, para cada una de las unidades que forman parte del contexto de un posible signo de fin de oración se asocia este vector junto con dos indicadores que expresan si la palabra en cuestión comienza por mayúscula y si sigue o no a un posible fin de oración. Los vectores correspondientes a las seis palabras de contexto conforman lo que se llama un patrón para la red neuronal. Este patrón (que son valores numéricos) se introduce como entrada a la red neuronal para que decida si el signo de puntuación es realmente un final de oración. Obviamente, para que la red pueda llevar a cabo esta tarea ha debido realizar previamente un período de aprendizaje. Es lo que se llama entrenamiento de la red neuronal.

3.4 Entrenamiento

Para proceder al entrenamiento de la red neuronal se prepararon dos conjuntos de patrones. El primero de ellos contenía ejemplos de casos de construcciones y estructuras del castellano, tanto usuales como casos especiales.

Con ello se pretendía que la red fuera capaz de identificar no sólo las estructuras que aparecen normalmente, sino también aquellas que pueden dar lugar a las ambigüedades que estamos tratando de resolver. En este conjunto se reunieron más de 600 patrones, aunque no se trató de que fuera un conjunto completo. Esto es, pretendíamos ver si con un conjunto de entrenamiento en cierto modo escaso, la red era capaz de generalizar y resolver ambigüedades que no hubiera visto anteriormente. El segundo conjunto de patrones era más pequeño. Era el llamado conjunto de validación cruzada, que nos permitió detener el entrenamiento en el punto en que la red había alcanzado el máximo grado de generalización para el conjunto de entrenamiento dado. Un último conjunto de patrones sería el formado por los patrones correspondientes a los textos en los que pretendemos resolver las posibles ambigüedades y separar en oraciones.

4 Evaluación

Para establecer la bondad del sistema realizamos diversas pruebas. Básicamente las pruebas consistían en entrenar la red neuronal con un conjunto de patrones de entrenamiento determinado y ver qué resultados obteníamos sobre los textos que tratábamos de separar en oraciones.

Los primeros conjuntos de entrenamiento fueron tomados con patrones escogidos al azar dentro del corpus. Los resultados no fueron todo lo buenos que se esperaban, dado que la red no podía clasificar bien patrones que no hubiera encontrado antes. Por esta razón modificamos el conjunto de entrenamiento para que encerrara casos en los que se daban ambigüedades. A medida que íbamos incrementando estos casos los resultados iban siendo mejores. Finalmente, con un conjunto de entrenamiento de más de 600 patrones escogidos con cierto cuidado (para que encerraran el mayor nĕmero posible de casos con ambigüedades), pero que en absoluto era exhaustivo, se obtuvieron resultados realmente buenos.

Comparamos nuestro método con otros dos que generamos para la ocasión. El primero de ellos es el método más sencillo que se nos puede ocurrir: consistía en un programa que buscaba los posibles finales de oración y separaba por ahí las oraciones. Este método marcaba el límite inferior, que en el caso del corpus del ABC quedó en un 95% aproximadamente. El segundo método fue una modificación de un segmentador de palabras creado en el departamento y que dio una eficacia ligeramente por encima del 97%. El sistema basado en la red neuronal, trabajando sobre el mismo conjunto de patrones alcanzó una efectividad del 97.5%.

Gracias al conjunto de validación cruzada llegamos a la conclusión de que era necesario escoger cuidadosamente los patrones para el entrenamiento. De otro modo, la red respondía erróneamente a los patrones que no había visto con anterioridad. Así pues, fuimos modificando el conjunto de entrenamiento hasta que, como se ha dicho en el párrafo previo, llegamos a un punto en el que, sin ser óptimos los resultados, se conseguía una generalización bastante aceptable.

Por otra parte, viendo los resultados sobre el conjunto de validación cruzada descubrimos que, como sugerían Palmer y Hearst, tan sólo hacían falta dos nodos en la capa intermedia. Con esos dos nodos se llegaba a un compromiso entre capacidad de generalización, capacidad de aprendizaje y velocidad para lograr dicho aprendizaje. Es decir, la red aprendía y generalizaba bien los patrones que se le presentaban a la entrada, sin llegar a convertirse en una simple

memoria de patrones. Además, lo conseguía de forma bastante rápida: tan sólo hicieron falta alrededor de 200 ciclos para obtener unos resultados apreciables.

También llegamos a la conclusión de que el contexto utilizado (tres tokens antes y después del posible límite de oración) era suficiente para que la red tuviera la información necesaria para clasificar correctamente los patrones.

En cuanto a los nodos que conforman la red neuronal, confirmamos las teorías que determinan que los nodos de las capas de entrada e intermedia deben tener funciones no lineales, bien sea la función de activación o bien la de salida. Por su parte, el nodo de la capa de salida debe ser lineal.

Es interesante resaltar también los buenos resultados que da la red *back-propagation*. Quizá se podría llegar a ellos mediante algún otro tipo de arquitectura de red supervisada, o incluso mediante una red no supervisada. Sin embargo, dadas las características del problema parece que el perceptrón multi-capa es la solución idónea. El algoritmo *back-propagation*, por su parte, proporciona un método seguro y rápido de convergencia.

5 Conclusiones

Este trabajo nació con la pretensión de desarrollar par el castellano un modelo semejante al que crearon Palmer y Hearst para el inglés. Estos dos autores habían utilizado una red neuronal para tratar de separar un texto en oraciones de forma efectiva.

Los resultados del método neuronal y del método adaptado del segmentador convencional, basado en detectar patrones utilizando expresiones regulares y tratar cada patrón mediante una regla, resultan bastante similares en la evaluación realizada. Sin embargo, cualquiera que haya desarrollado un segmentador sabe que, en la mayoría de los casos estos, este será útil solamente para un determinado conjunto de textos que reúnen una serie de características. Por ejemplo, nuestro segmentador basado en reglas escrito específicamente para tratar los textos publicados en el suplemento cultural del diario ABC, es poco menos que inservible para la misma tarea cuando tratamos de aplicarlo a la colección de textos de la Unión Internacional de Telecomunicaciones.

Esta hecho establece una diferencia sustancial entre el método aquí descrito, basado en el empleo de una red neuronal, y los métodos convencionales: si bien en éstos habría que dedicar un gran esfuerzo a confeccionar las nuevas reglas que permitieran separar correctamente las oraciones en las que el segmentador falla, aquí tan sólo habría que tomar dichas oraciones y presentárselas como ejemplo a la red.

Referencias

[1] G.E. Hinton. *Connectionist Symbol Processing.* The MIT Press, 1991.

[2] Don R. Hush and Bill G. Horne. Progress in supervised neural networks. What's new since Lippman? *IEEE Signal Processing Magazine*, pages 8–39, January 1993.

[3] David D. Palmer. Satz - an adaptive sentence segmentation system. Technical Report UCB/CSD-94-797, Computer Science Division, University of California, Berkeley, 1994.

[4] David D. Palmer and Marti A. Hearst. Adaptive sentence boundary disambiguation. In *Proceedings of the Fourth Conference on Applied Natural Language Processing*, Stuttgart, Germany, 1995. (Also available as Technical Report UCB/CSD-94-797 of the Computer Science Division, University of California, Berkeley).

[5] Ronan G. Reilly and Noel E. Sharkey, editors. *Connectionist Approaches to Natural Language Processing.* Lawrence Erlbaum, 1992.

[6] Fernando Sánchez and Amalio Nieto. Desarrollo de un etiquetador morfosintáctico para el español. In *Actas del XI Congreso de la Sociedad Española para el Procesamiento del Lenguaje Natural (SEPLN'95)*, Bilbao, September 1995.

[7] Stefan Wermter. *Hybrid Connectionist Natural Language Processing.* Chapman & Hall, 1995.

[8] Stefan Wermter, Ellen Riloff, and Gabriele Scheler, editors. *Connectionist, Statistical, and Symbolic Approaches to Learning for Natural Language Processing.* Lecture Notes in Artificial Intelligence. Springer-Verlag, 1996.